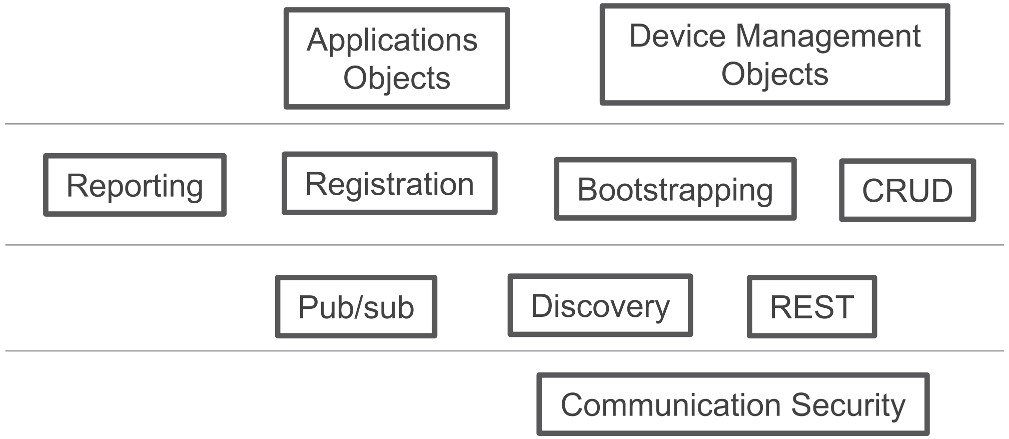
IOT DEVICE MANAGEMENT

Jaime Jiménez
jaime.jimenez@ericsson.com

INTERNET PROTOCOLS TO THE EDGE



Applications Objects

Device Management Objects

Reporting

Registration

Bootstrapping

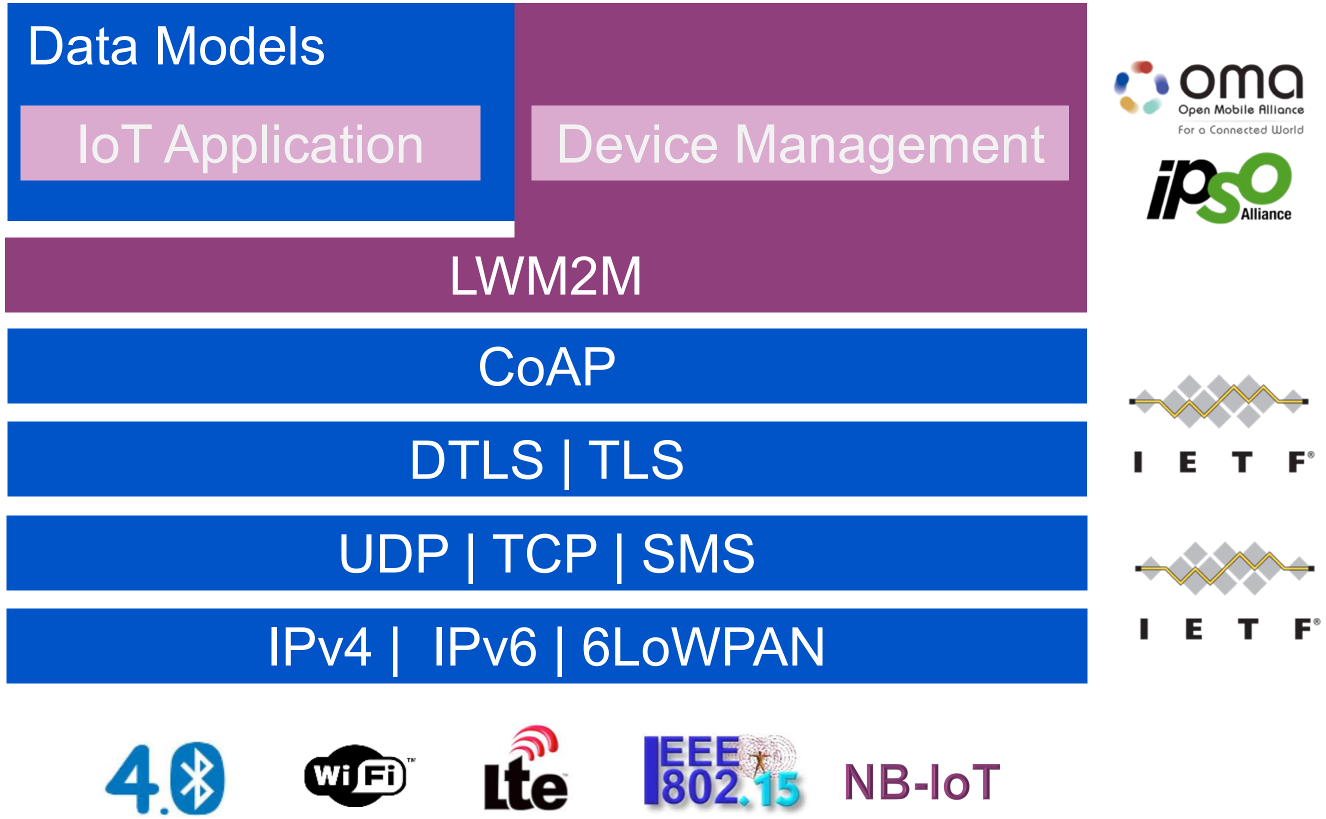
CRUD

Pub/sub

Discovery

REST

Communication Security



Data Models

IoT Application

Device Management

LWM2M

CoAP

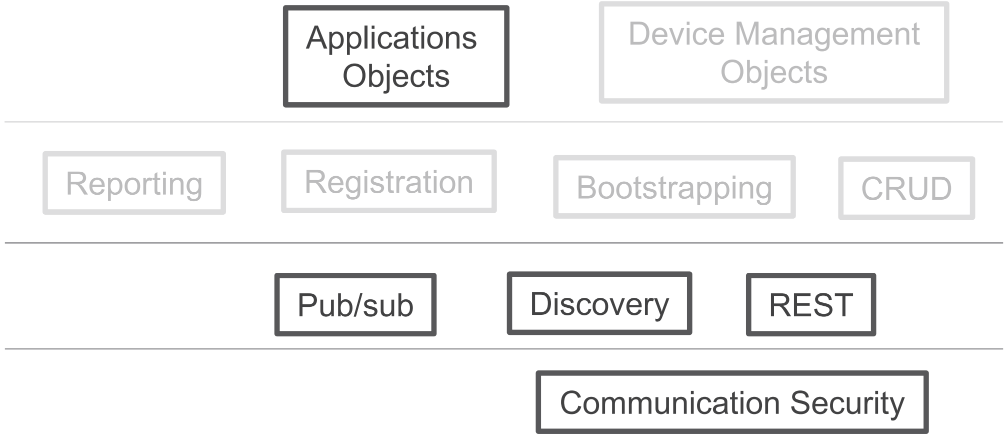
DTLS | TLS

UDP | TCP | SMS

IPv4 | IPv6 | 6LoWPAN

- In order to use the Web and the Internet to the best possible extend IoT devices need to support IP.
- Non-standard approaches are a risk
  - Particularly when it comes to rolling out your own, custom security mechanism.

ONLY COAP



Applications Objects

Device Management Objects

Reporting

Registration

Bootstrapping

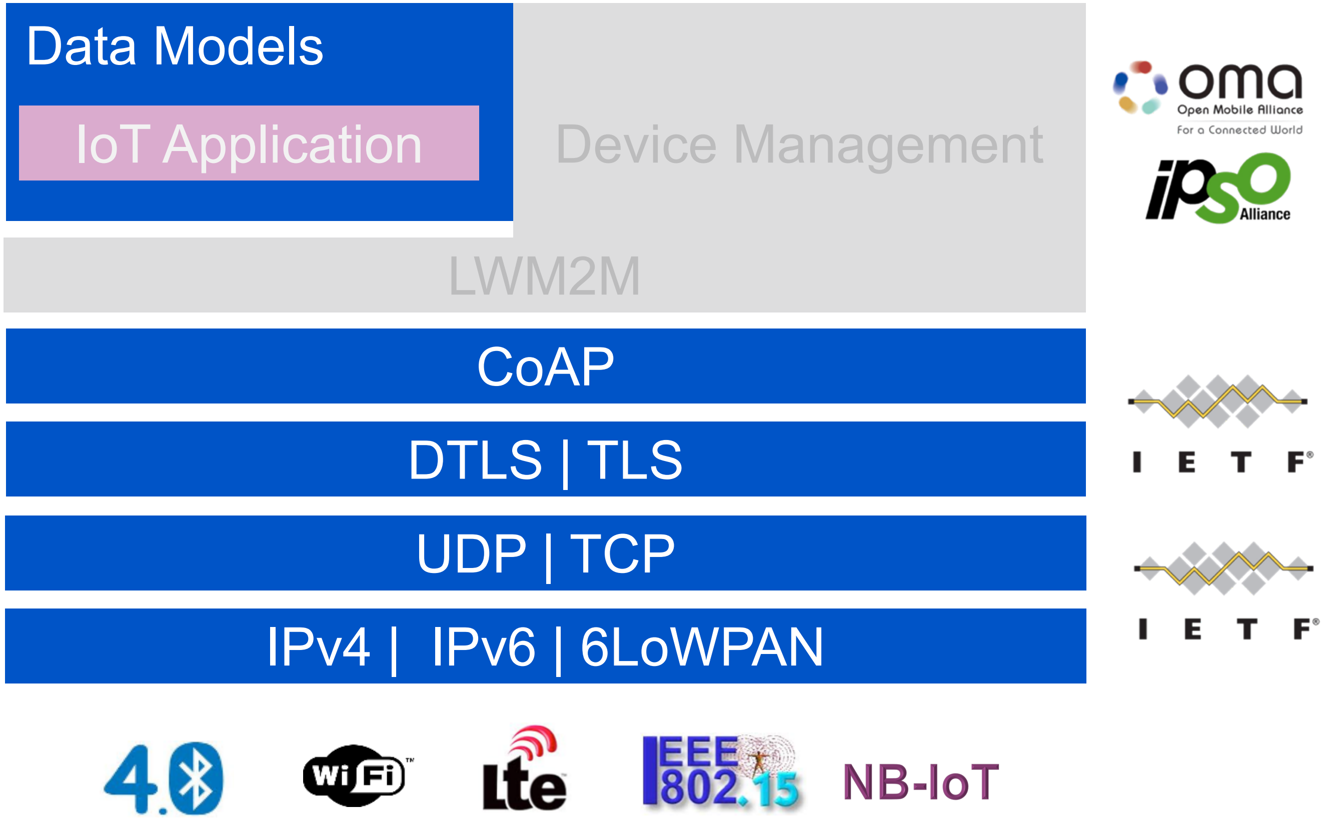
CRUD

Pub/sub

Discovery

REST

Communication Security



Data Models

IoT Application

Device Management

LWM2M

CoAP

DTLS | TLS

UDP | TCP

IPv4 | IPv6 | 6LoWPAN

oma Open Mobile Alliance For a Connected World

ipso Alliance

IETF

IETF

4.0

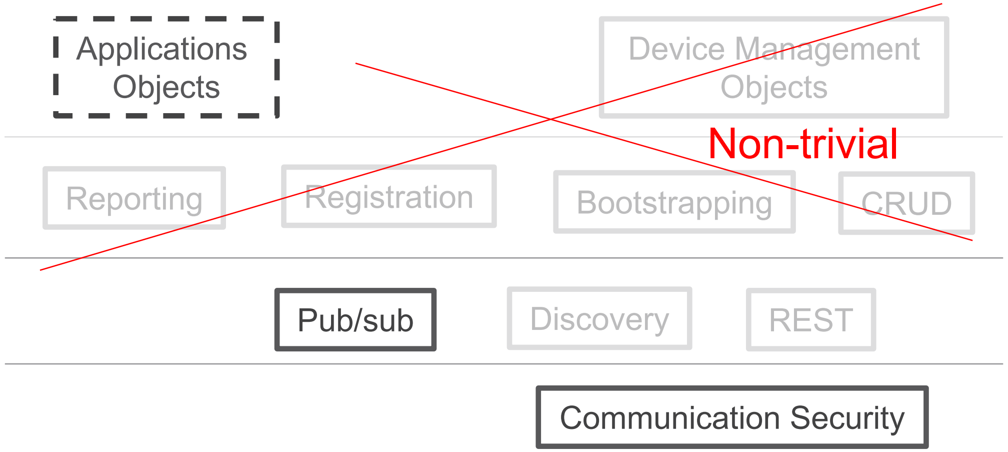
WiFi

lte

IEEE 802.15

NB-IoT

ONLY MQTT



Applications Objects

Device Management Objects

Non-trivial

Reporting

Registration

Bootstrapping

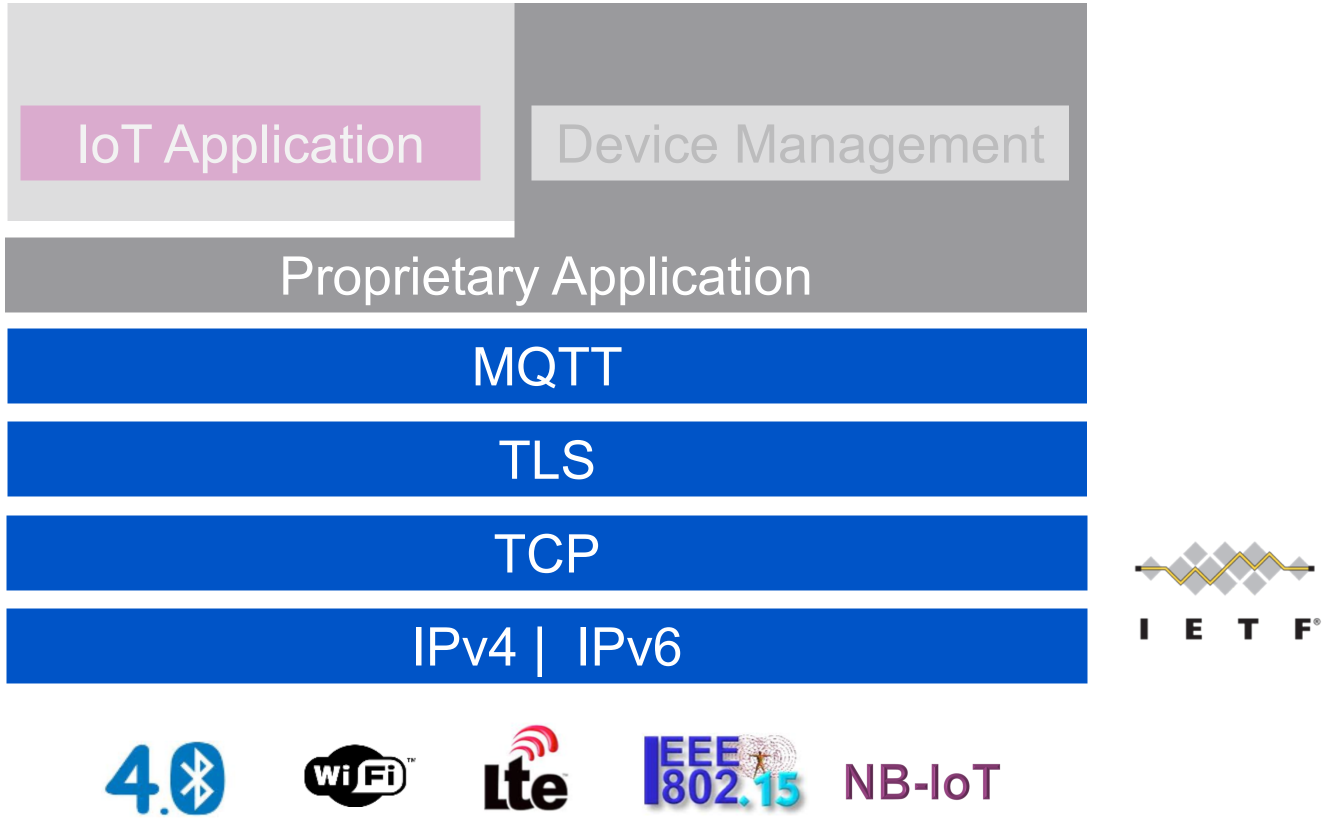
CRUD

Pub/sub

Discovery

REST

Communication Security



IoT Application

Device Management

Proprietary Application

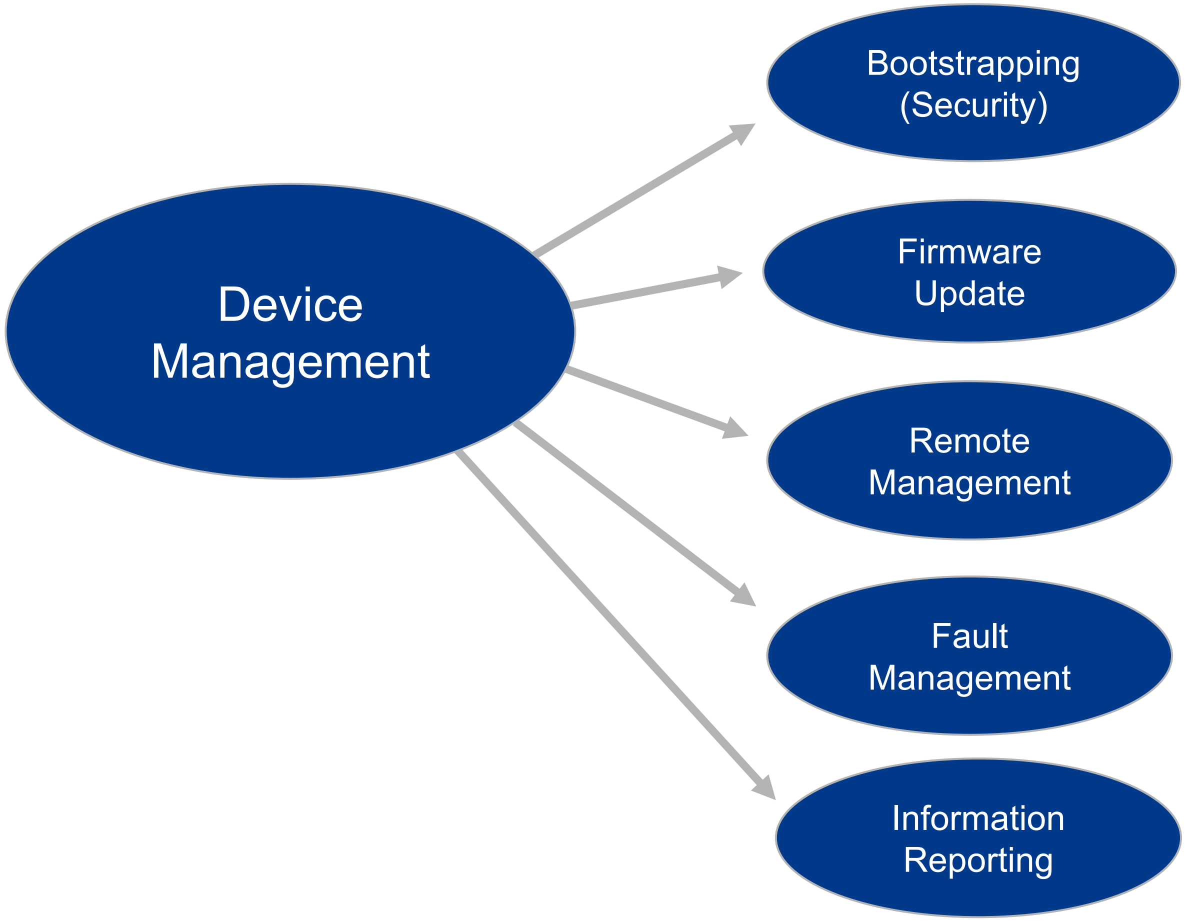
MQTT

TLS

TCP

IPv4 | IPv6

DIVERSE IOT DEPLOYMENTS WITH COMMON NEEDS



- Service provisioning
- Key management
- Provisioning of access control lists

- Update application and system software
- Apply bug fixes and add new features

- Changes to settings
- Trigger actuators

- Report errors from devices
- Query status of devices

- Notify changes in sensor values
- Retrieve configuration settings and device status

LWM2M 1.0 ARCHITECTURE

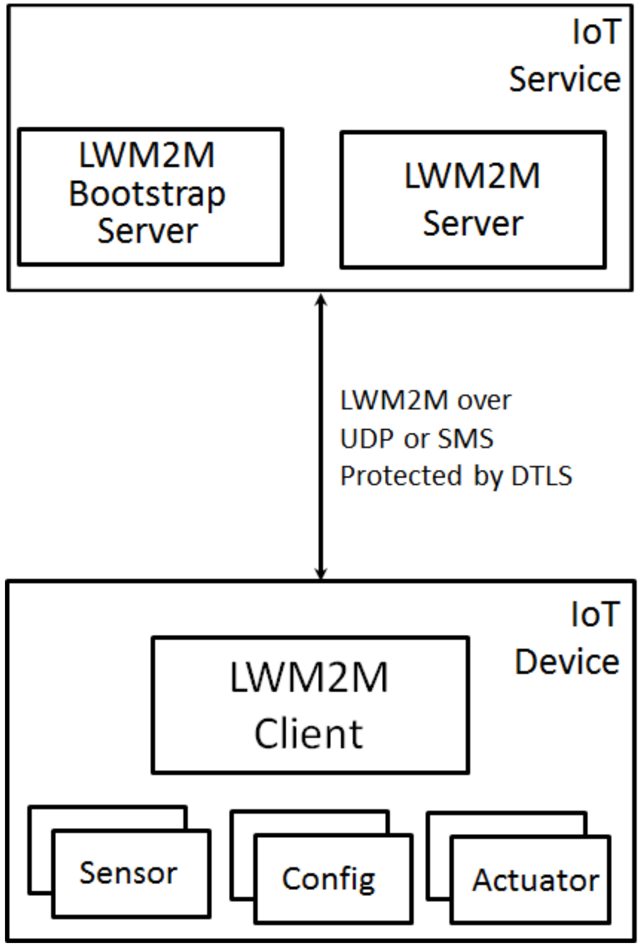


Figure 1: Entities in the LWM2M Architecture.

LightweightM2M
specified at oma

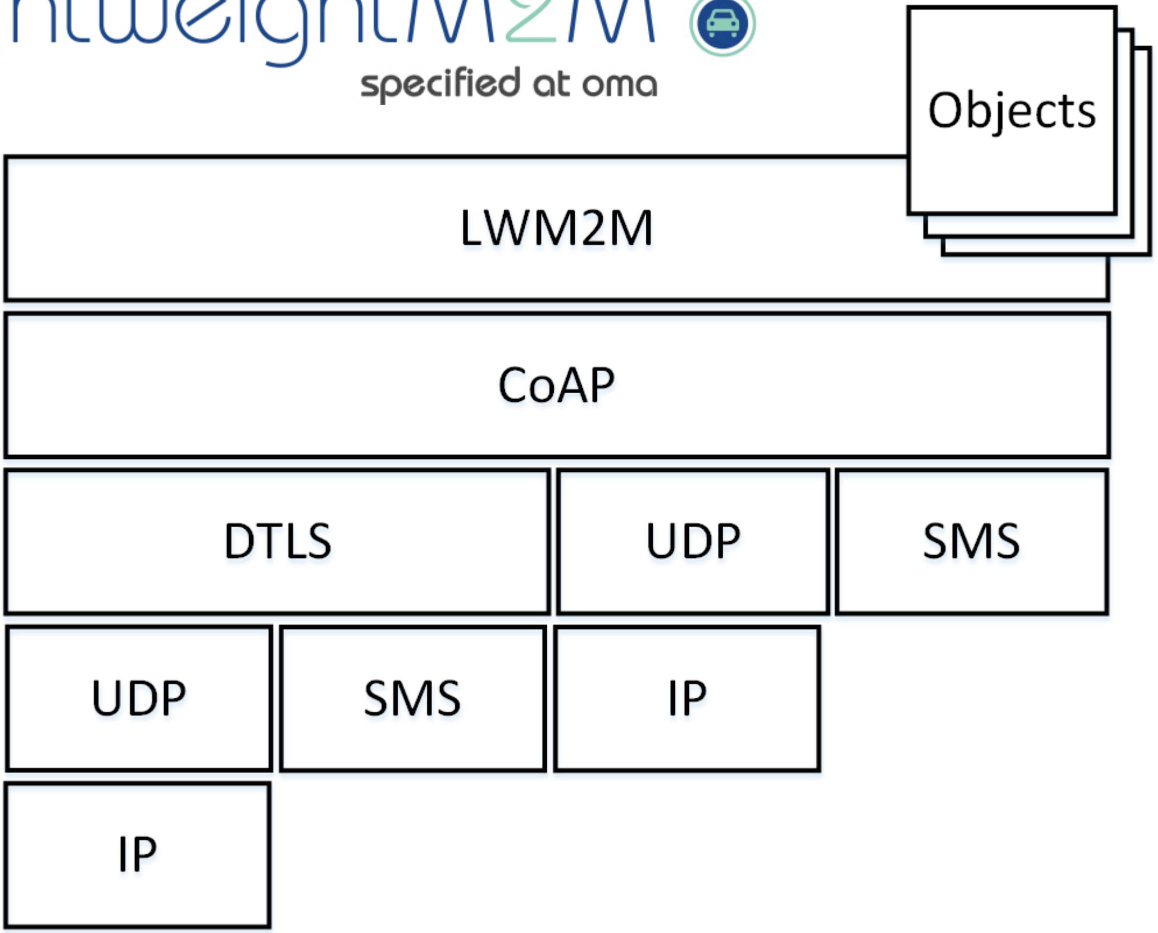
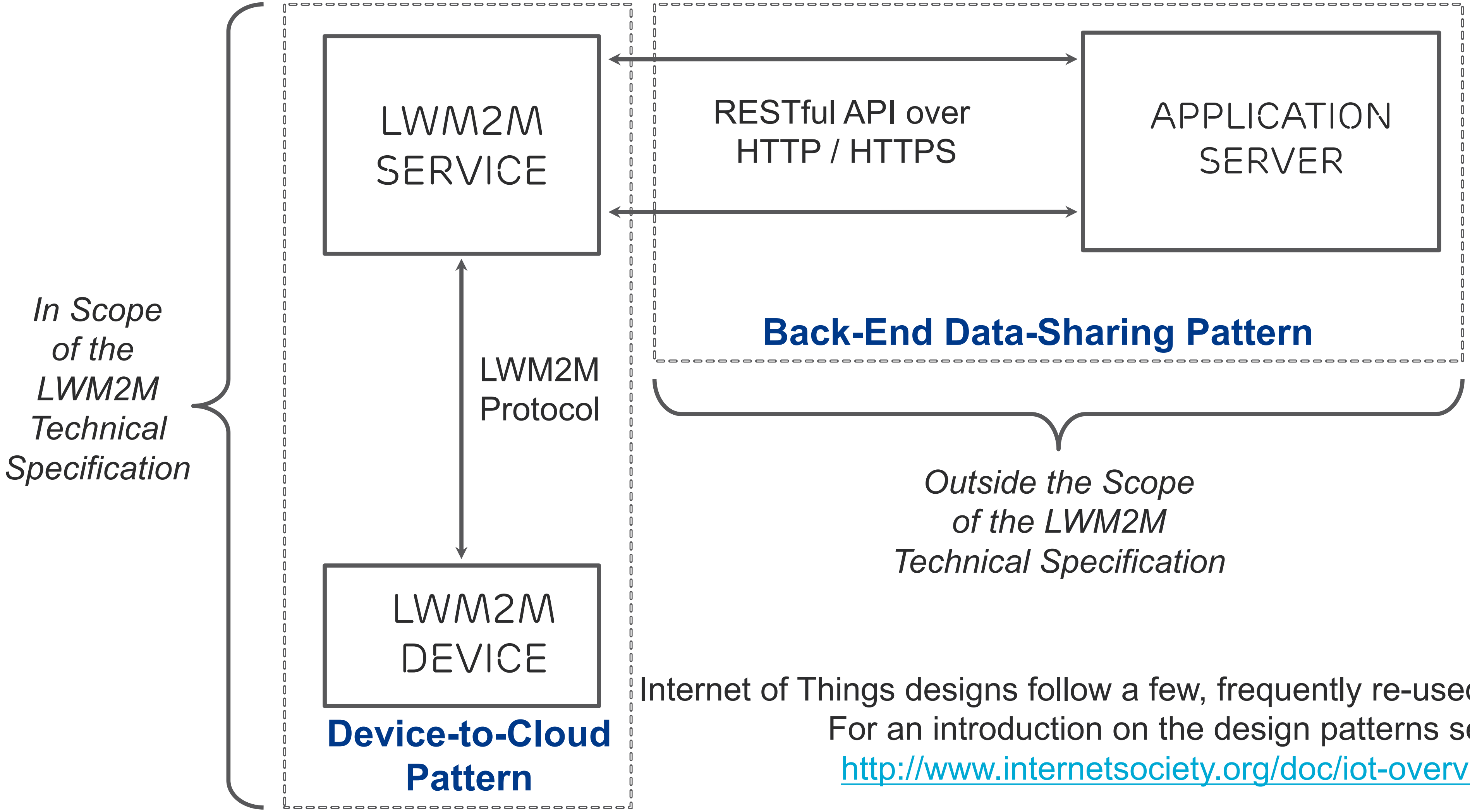


Figure 2: Protocol Stack

LWM2M 1.0 ARCHITECTURE



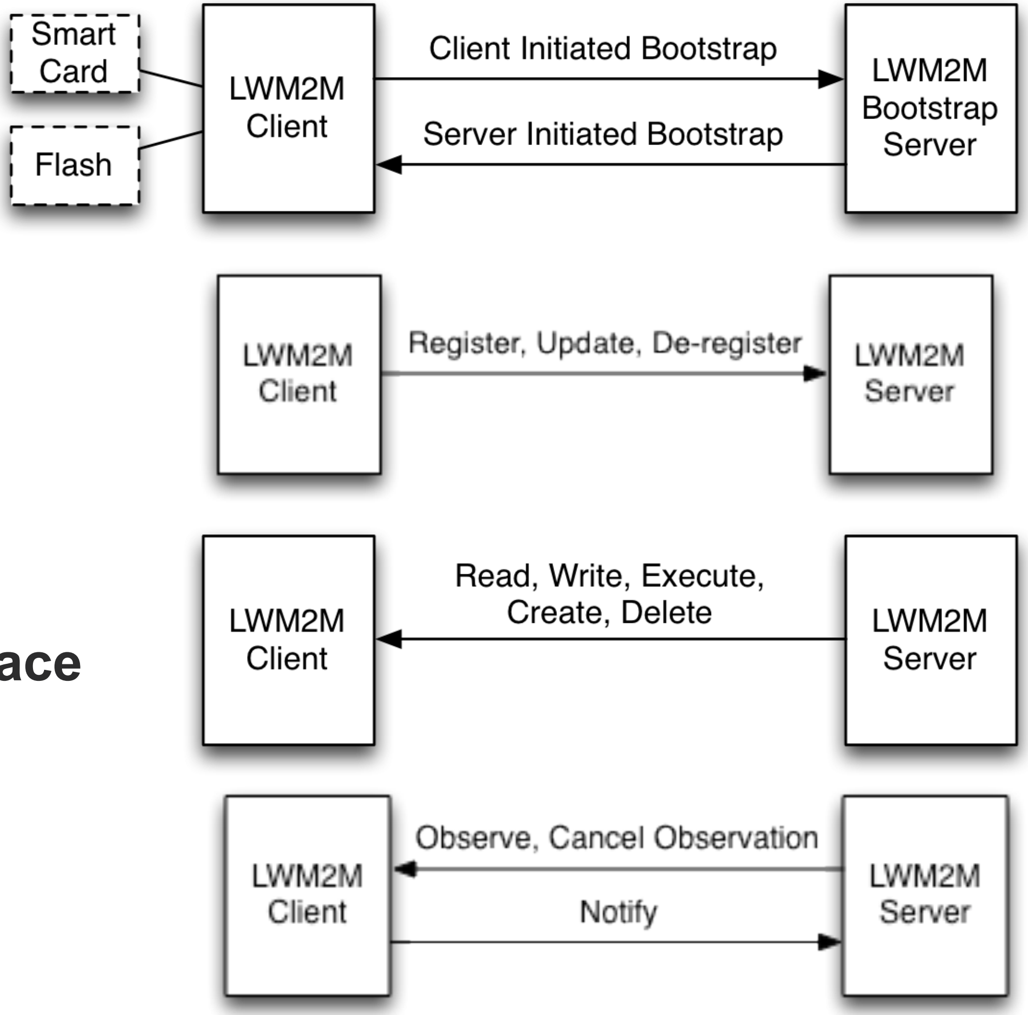
In Scope of the LWM2M Technical Specification
LWM2M SERVICE
RESTful API over HTTP / HTTPS
APPLICATION SERVER
Back-End Data-Sharing Pattern
LWM2M Protocol
Outside the Scope of the LWM2M Technical Specification
LWM2M DEVICE
Device-to-Cloud Pattern

Internet of Things designs follow a few, frequently re-used design patterns. For an introduction on the design patterns see at http://www.internetsociety.org/doc/iot-overview.

THE LWM2M RESTFUL API

High-level message pattern hiding details of networking and security protocols

- **Bootstrap interface**
  - Configure servers info, credentials & ACLs
- **Registration interface**
  - Informs server about “existence” and supported functionality (e.g., objects, transport bindings)
- **Device management & service enablement interface**
  - Ability to access object instances and resources
- **Information reporting interface**
  - Publish/subscribe interaction for observing changes in resources.



Smart Card
Flash
LWM2M Client
Client Initiated Bootstrap
Server Initiated Bootstrap
LWM2M Bootstrap Server
LWM2M Client
Register, Update, De-register
LWM2M Server
LWM2M Client
Read, Write, Execute, Create, Delete
LWM2M Server
LWM2M Client
Observe, Cancel Observation
Notify
LWM2M Server

BUILDING BLOCKS FOR LWM2M VERSION 1.0

COAP

- Specified in RFC 7252, uses UDP.
- Short, binary header.
- Publish/Subscribe support with RFC 7641.
- Designed for small data transmissions but capable of transferring large data as well with RFC 7959.
- Reliable transport support with RFC 8323.
- Built-in support for discovery.
- Lots of open source implementations available.

DTLS

- Specified in RFC 6347 and builds on TLS 1.2
- Offers communication security by providing confidentiality, integrity and authentication.
- Performance depends on selected ciphersuite and settings.
- The full list of standardized ciphersuites can be found at IANA.

Object Model

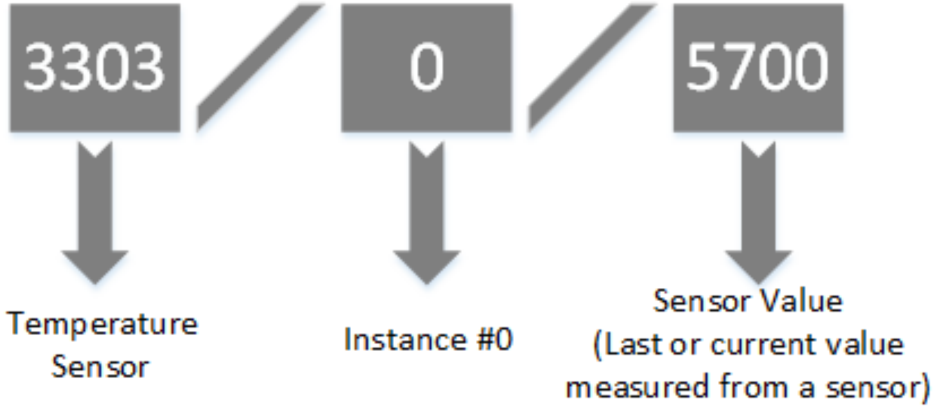
- Objects add functionality for device management, security foundation and applications.
- Reusable resources allow for flexible applications.
- Multiple serialization options.
- Better compression (CBOR)
- Large number of objects defined and listed in repository.

OBJECT MODEL

Objects/Resources are accessed with simple URIs:

```
/{Object ID}/{Object Instance}/{Resource ID}
```

Example:

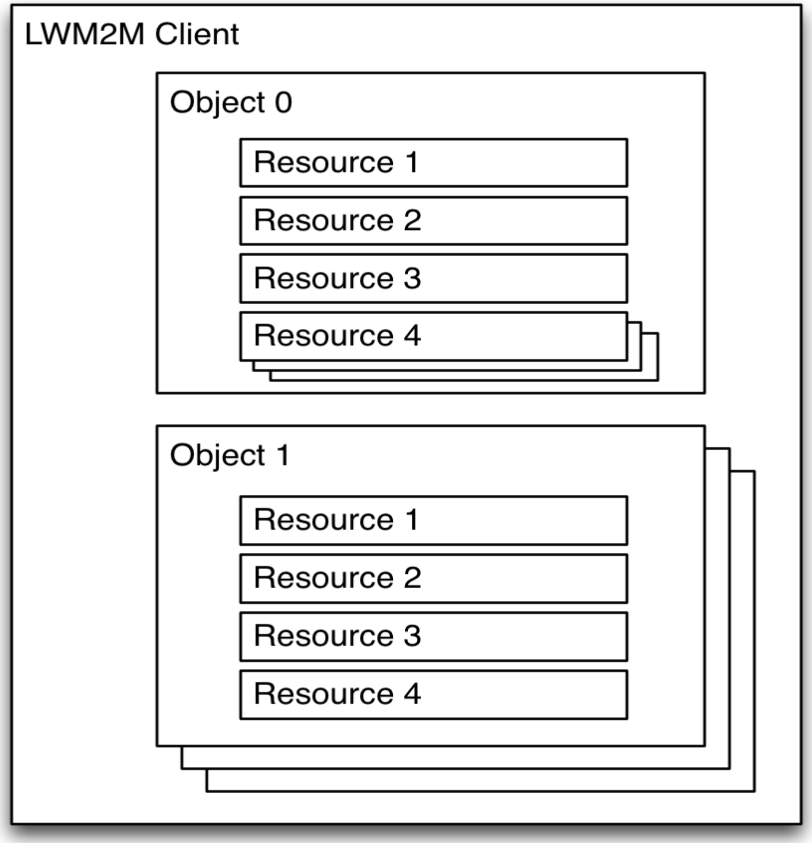


3303 / 0 / 5700

Temperature Sensor

Instance #0

Sensor Value (Last or current value measured from a sensor)



LWM2M Client

Object 0

Resource 1

Resource 2

Resource 3

Resource 4

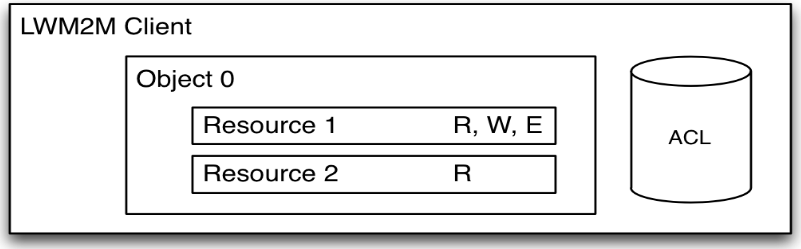
Object 1

Resource 1

Resource 2

Resource 3

Resource 4



LWM2M Client

Object 0

Resource 1	R, W, E
Resource 2	R

ACL

OBJECTS

The LWM2M technical specification itself defines eight objects; the repository contains many more contributed by IPSO alliance, oneM2M, and from vendors.

LWM2M Security	0	Keying material of a LWM2M Client to access a LWM2M server.
LWM2M Server	1	Data related to a LWM2M server.
Access Control	2	Information used to check whether a LWM2M Server has access to object.
Device	3	Device related information, including device reboot and factory reset function.
Connectivity Monitoring	4	Parameters related to network connectivity.
Firmware	5	Capability to update firmware
Location	6	Device location information
Connectivity Statistics	7	Information like transmit and receive counters

EXAMPLE: IPSO TEMPERATURE OBJECT

		Operations	Type	
Sensor value	5700	R	Float	Last or current measured value from the sensor
Min measured value	5601	R	Float	The minimum value measured by the sensor since power ON or reset
Max measured value	5602	R	Float	The maximum value measured by the sensor since power ON or reset
Min range value	5603	R	Float	The minimum value that can be measured by the sensor
Min range value	5604	R	Float	The minimum value that can be measured by the sensor
Max range value	5604	R	Float	The maximum value that can be measured by the sensor
Sensor units	5701	R	String	Measurement units definition
Reset min and max measured values	5605	E	String	Reset the min and max measured values to current value

Data

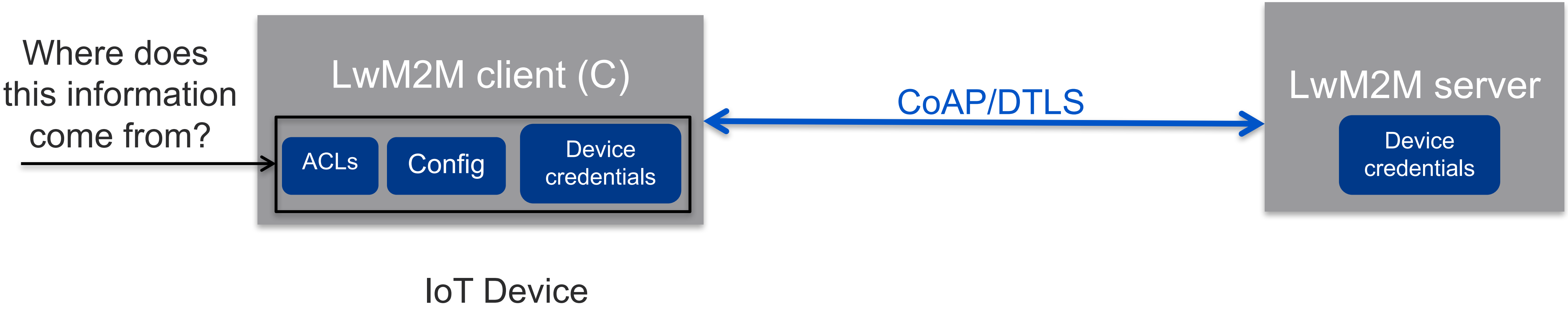
Metadata

Actions

A detailed description of this object and many others can be found at the IPSO Github repository.

BOOTSTRAPPING ARCHITECTURE

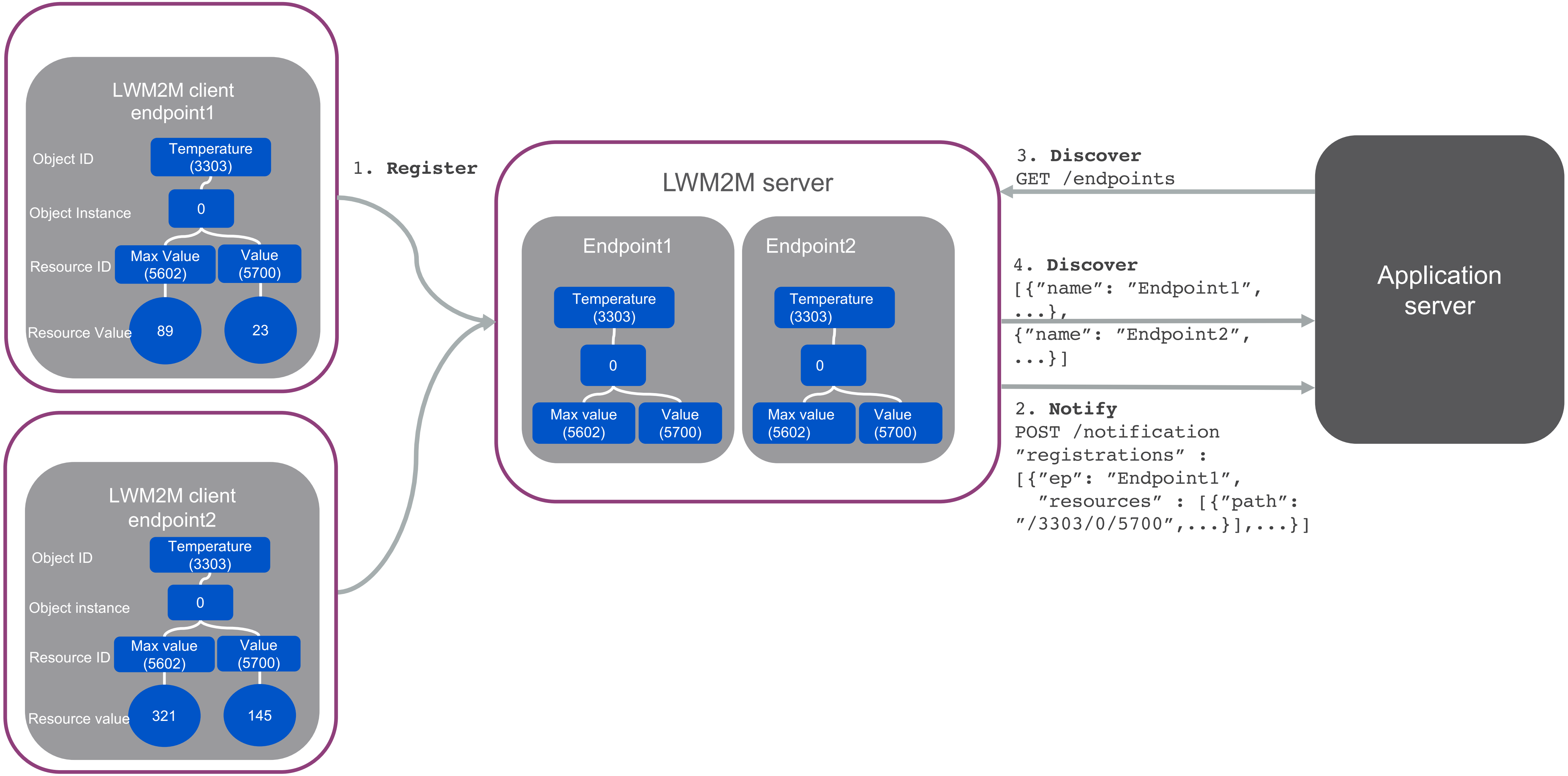
- LwM2M client needs credentials to securely communicate with the LwM2M server using DTLS. Configuration and access rights might change.



Specification of several deployment choices:

- Factory bootstrap
- Bootstrap from smartcard
- Client initiated bootstrap
- Server initiated bootstrap

DEVICE DISCOVERY



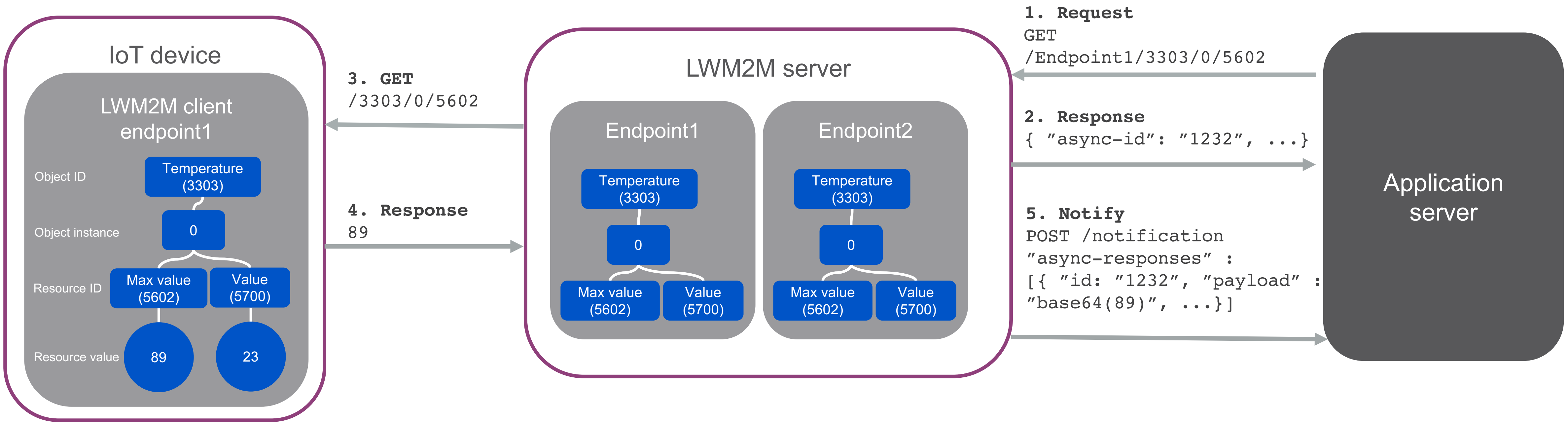
1. **Register**

3. **Discover**
```
GET /endpoints
```

4. **Discover**
```
[{"name": "Endpoint1",
...},
{"name": "Endpoint2",
...}]
```

2. **Notify**
```
POST /notification
"registrations" :
[{"ep": "Endpoint1",
  "resources" : [{"path":
"/3303/0/5700",...}],...}]
```

INFORMATION RETRIEVAL



```
1. Request
GET
/Endpoint1/3303/0/5602
```

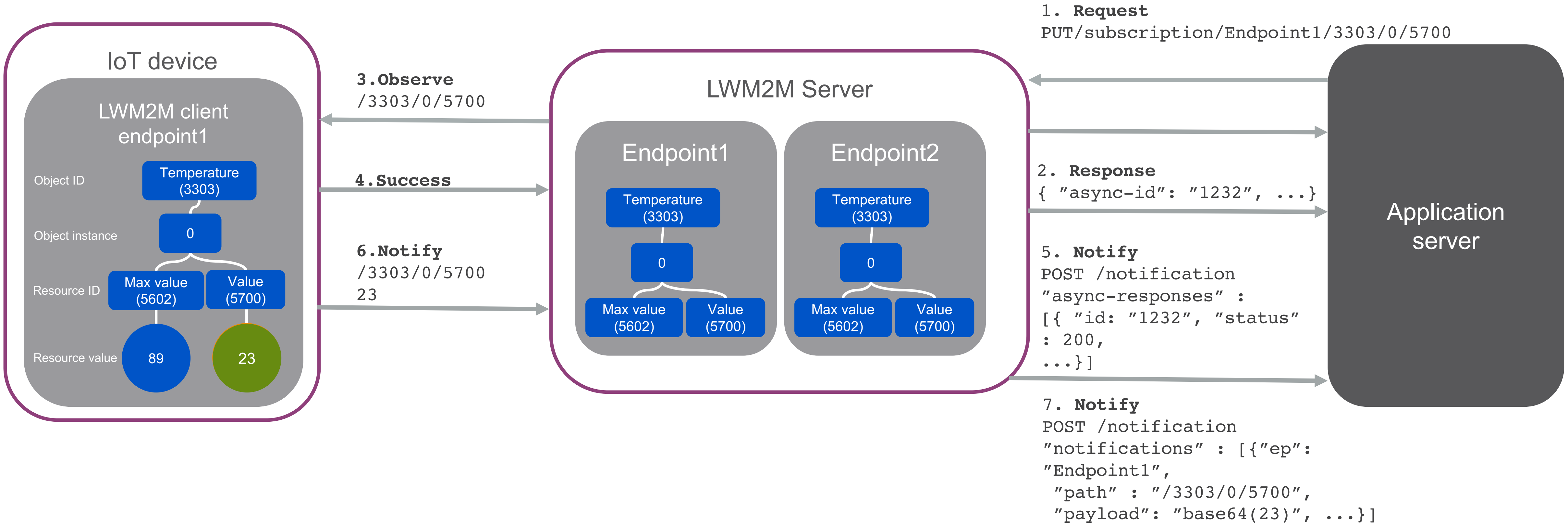
```
2. Response
{ ”async-id”: ”1232”, ...}
```

```
3. GET
/3303/0/5602
```

```
4. Response
89
```

```
5. Notify
POST /notification
”async-responses” :
[{ ”id: ”1232”, ”payload” :
”base64(89)”, ...}]
```

INFORMATION REPORTING



```
1. Request
PUT/subscription/Endpoint1/3303/0/5700
```

```
2. Response
{ ”async-id”: ”1232”, ...}
```

```
3.Observe
/3303/0/5700
```

```
4.Success
```

```
5. Notify
POST /notification
”async-responses” :
[{ ”id: ”1232”, ”status”
: 200,
...}]
```

```
6.Notify
/3303/0/5700
23
```

```
7. Notify
POST /notification
”notifications” : [{”ep”:
”Endpoint1”,
 ”path” : ”/3303/0/5700”,
 ”payload”: ”base64(23)”, ...}]
```

OUTLOOK

Version 1.0 has been published in Feb. 2017 following many interop-events.

Work on version 1.1 is about to conclude. Continuous interop testing

Additional functionality:

- Support for CoAP over TCP/TLS
- Protocol gateway support
- Better compression, serialization with SENML, CBOR)
- More security features, such as DTLS IoT profile compliance, and application layer security.
- Various performance optimizations

LWM2M: HOW TO PARTICIPATE?

- I want to contribute to the technical specification
  - Submit new objects to the repository.
  - File issues with the public OMA LWM2M Github issue tracker.
  - Become OMA member and participate in the standardization process.
  - Participate in the IETF for foundational standards (such as CoAP, CBOR, DTLS/TLS, HTTP, etc.)
- I want to write code
  - Several open source projects are happy to received your contributions.
  - Examples: coap.technology, Leshan and Wakaama
- I want to test my implementation
  - Join an interoperability test event (PlugFest, TestFest). Info about upcoming events can be found at the OMA testfest website.
  - Use one of the available open source implementations to test against.